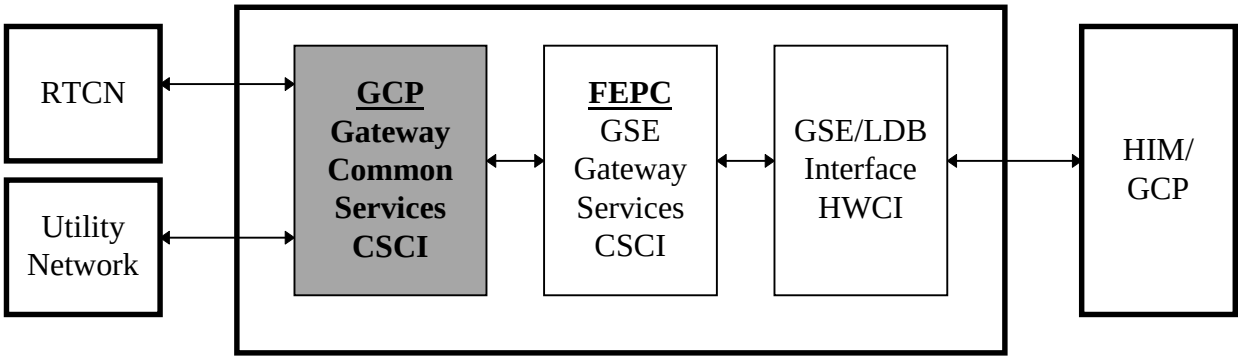
1. Gateway Common Services CSCI

1.1 Gateway Common Services CSCI Introduction

1.1.1 Gateway Common Services CSCI Overview

The Gateway Common Services CSCI provides the essential functions to make any CLCS gateway operational. It is resident in the Gateway Control Processor (GCP). The Gateway Common Services CSCI is composed of multiple concurrent tasks that perform individual functions in order to support all the resources in the Gateway.



1.1.2 Gateway Common Services CSCI Operational Description

The Gateway Common Services CSCI is initiated by the Real Time Operating System (RTOS) resident on the Gateway's local disk. Initially it will spawn all the necessary tasks to support the Gateway services.

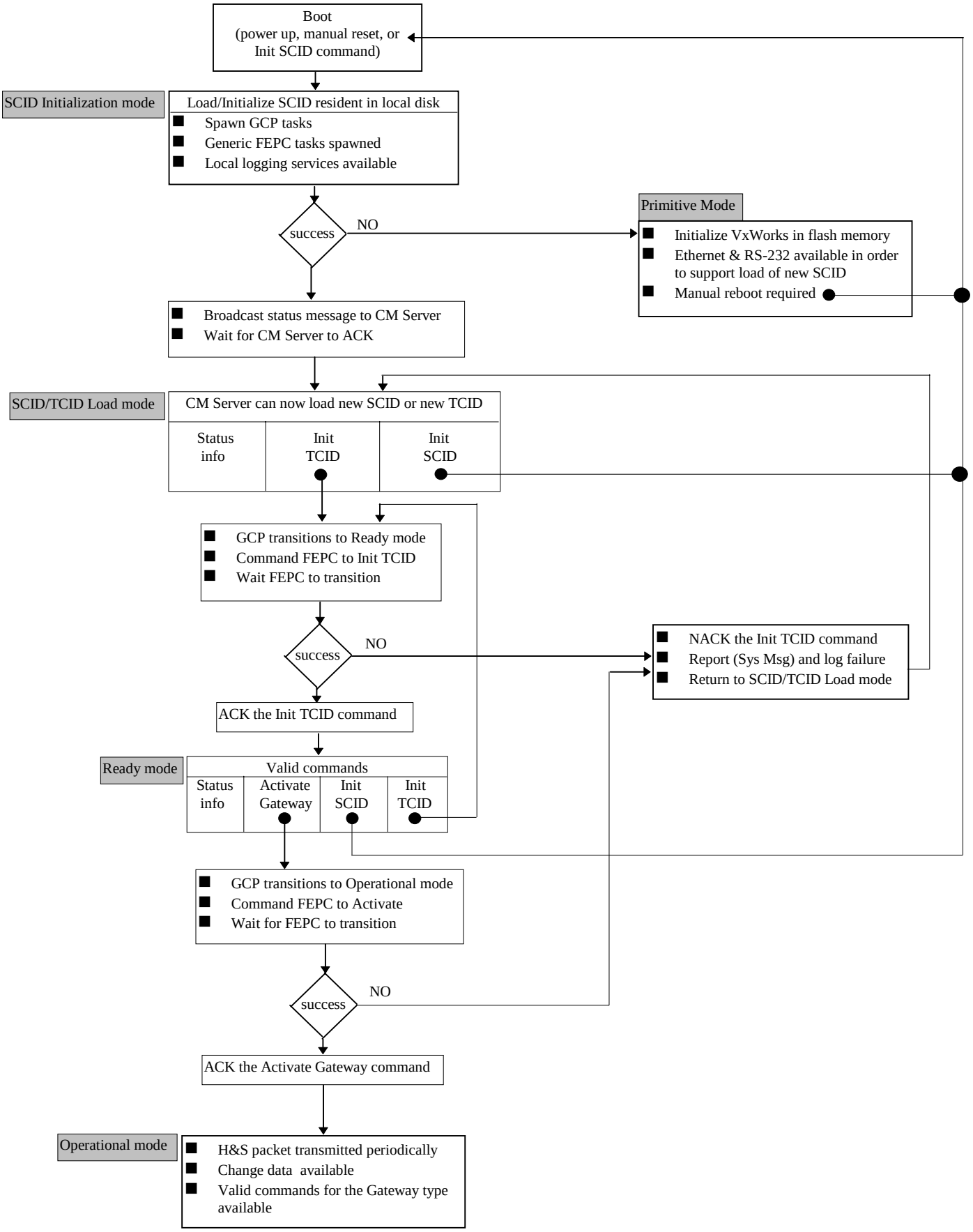
Commands are received asynchronously over the RTCN. Commands may be one of several types (e.g. Table Maintenance, Load / Initialize, end item commands (Data Bus), etc.). In all cases a response is returned to the issuer of the command. End item commands are routed to the FEPC command queue. The Redstone delivery will support only Data Bus and Load / Initialization commands over the RTCN.

1.1.3 Gateway Common Services CSCI Operating Modes

The Gateway Common Services will transition through four (4) normal modes before being fully operational (SCID Initialization, SCID/TCID Load, Ready, and Operational). In case of an error during boot, the Gateway will transition to a Primitive mode in order to support re-loading of the SCID via the Utility Network (ethernet).

- Gateway Common Services operating modes:
  - SCID Initialization mode (boot or reboot)
    - The Gateway enters this mode via power-up, manual reset, or by CLCS command (Init SCID command).
    - VxWorks Boot Software (in FLASH) loads SCID from local hard disk.
    - Gateway Common Services CSCI initializes and spawns appropriate tasks on GCP.
    - The RTCN Interface is initialized at this time.
    - FEPC Spawns Init task. No Gateway specific FEPC software is operating at this time (i.e. no software to communicate to a GDB interface hardware or to process End Item Commands).
    - If SCID initialization was successful, Gateway Common Services CSCI broadcasts status message to CM Server indicating successful SCID Initialization. If the SCID initialization fails, error messages will be logged describing the boot errors (if possible). At the point the Gateway will attempt to boot the VxWorks operating system loaded into flash and will enter 'Primitive' mode.
    - CM Server acknowledges periodic status message. The acknowledgment instructs Gateway to enter 'SCID/TCID Load' mode.

- Primitive mode
  - This is not a normal state for the Gateway. If the SCID initialization fails, the Gateway will attempt to boot the minimal VxWorks operating system loaded into flash and will enter the 'Primitive' mode. SCID initialization failure could result if the SCID file transfer was corrupted or the hard disk was not accessible.
  - The Gateway will only be accessible via a Telnet or FTP session across the Gateway Group Utility Network or via the console port (RS-232) and a terminal.
  - The RTCN interface will not be initialized and will not be functional.
  - The Gateway will not accept CLCS Commands.
  - The Gateway Status information will not be available via the RTCN.
  - This state is intended to allow a new SCID to be loaded manually on the local hard disk via an FTP session across the Gateway Group Utility Network. A manual re-boot will be executed and the Gateway will attempt to enter the SCID Initialization mode.
- SCID/TCID Load mode
  - After receipt of the SCID Initialization Acknowledge from the CM Server, the Gateway enters the SCID/TCID Load mode. This is the normal mode when the Gateway is not in the 'Ready' mode or 'Operational' mode.
  - At this point, the Gateway is ready to receive normal Commands from the CM Server and to provide status information upon request via the RTCN.
  - 'End Item' commands will not be accepted.
  - This mode is intended to allow the Gateway to be polled for Status via the RTCN and to allow a new SCID or TCID to be loaded from the CM server to the Gateway's local hard disk via FTP 'put'.
  - SCID or TCID loads will verified by the CM Server via FTP 'get'.
  - Other than requests for status, the Gateway would normally receive an 'Init TCID' or 'Init SCID' command.
  - The 'Init SCID' command instructs the Gateway to re-boot and to enter the 'SCID Initialization Mode'.
  - The 'Init TCID' command instructs the GCP and the FEPC to load TCID tables off the local hard disk.
  - If TCID Initialization is successful, the Gateway transmits status to the CM server and enters the 'Ready' mode.
  - If TCID Initialization fails, the Gateway transmits status to the CM server and stays in the 'SCID/TCID Load' mode.
- Ready mode
  - At this point the Gateway successfully booted from the SCID and initialized TCID tables.
  - The Gateway will continue to process a very limited set of commands: requests for status, 'Init SCID', 'Init TCID' and 'Activate Gateway'.
  - 'End Item' commands will not be processed.
  - If the Init SCID command is received, the Gateway will enter the 'SCID Initialization' mode
  - If the Init TCID command is received, the GCP will instruct the FEPC to initialize TCID.
  - Normally the next command expected to be received is 'Activate Gateway'. The Activate Gateway command instructs the Gateway to spawn all Gateway specific tasks on the FEPC required for normal operation.
  - Upon successful initialization of the FEPC tasks, the Gateway transmits status to the CM server and enters the 'Operational' mode.
  - If FEPC task initialization was not successful, the Gateway transmits appropriate status to the CM server and returns to the 'SCID/TCID Load' mode.
- Operational mode
  - At this point, the Gateway will process all normal Commands.
  - Typically, the 'Activate Data Acquisition Command' would be sent to the Gateway by the CCP to begin processing data.
  - The 'Inhibit Data Acquisition' Command is used to terminate Gateway measurement processing.
  - 'Terminate Gateway' Command instructs the Gateway to transition back to 'Ready Mode'.

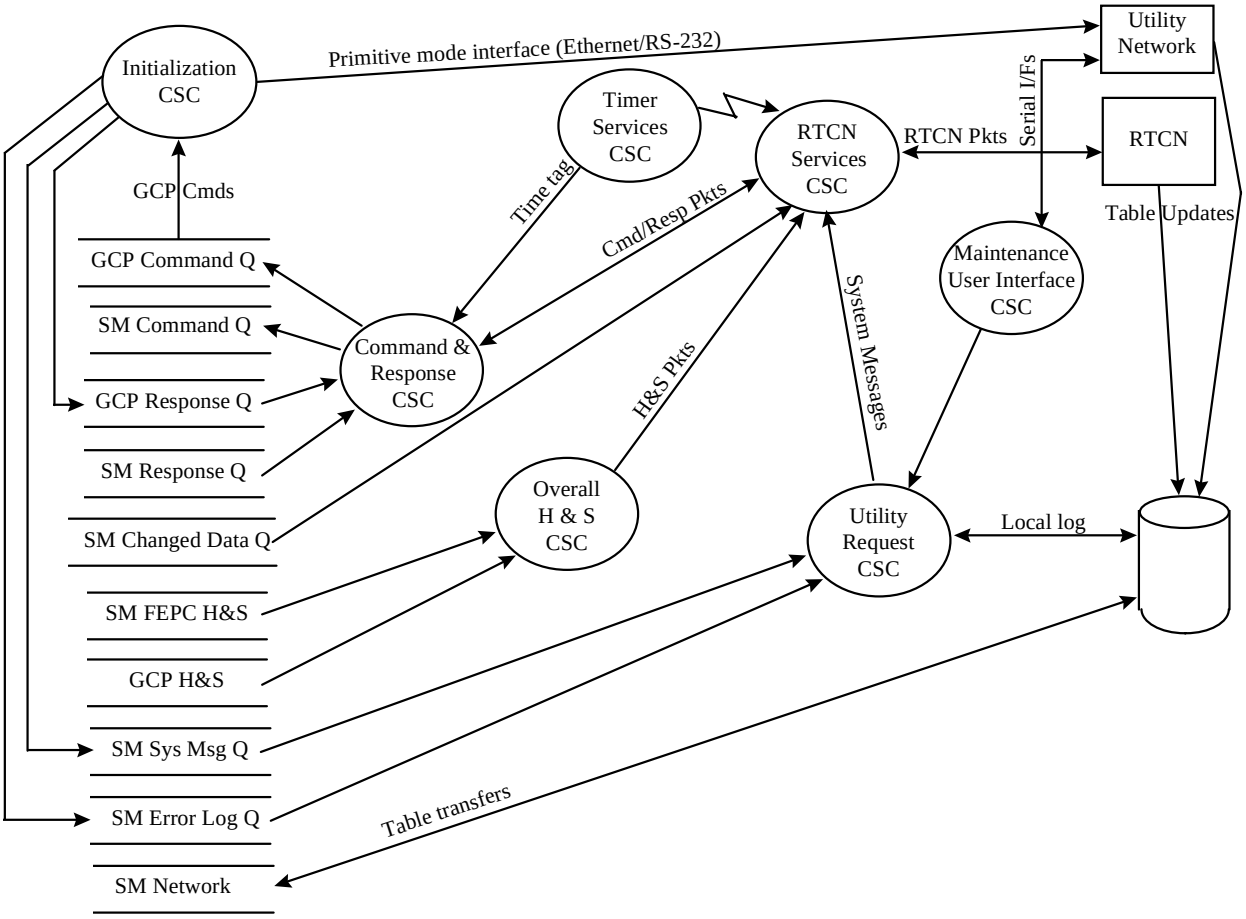

Boot
(power up, manual reset, or
Init SCID command)
SCID Initialization mode
Load/Initialize SCID resident in local disk
Spawn GCP tasks
Generic FEPC tasks spawned
Local logging services available
success
NO
Primitive Mode
Initialize VxWorks in flash memory
Ethernet & RS-232 available in order to support load of new SCID
Manual reboot required
Broadcast status message to CM Server
Wait for CM Server to ACK
SCID/TCID Load mode
CM Server can now load new SCID or new TCID
Status info
Init TCID
Init SCID
GCP transitions to Ready mode
Command FEPC to Init TCID
Wait FEPC to transition
success
NO
NACK the Init TCID command
Report (Sys Msg) and log failure
Return to SCID/TCID Load mode
ACK the Init TCID command
Ready mode
Valid commands
Status info
Activate Gateway
Init SCID
Init TCID
GCP transitions to Operational mode
Command FEPC to Activate
Wait for FEPC to transition
success
NO
ACK the Activate Gateway command
Operational mode
H&S packet transmitted periodically
Change data available
Valid commands for the Gateway type available

1.1.4 Gateway Common Services CSCI Computer Software Components

Gateway Common Services CSCI is composed of the following CSCs:

- Gateway Initialization CSC
- Gateway Command and Response CSC
- Gateway Table load, Initialization, and Maintenance CSC (removed)
- Gateway RTCN Services CSC
- Gateway Timer Services CSC
- Gateway Overall Health & Status CSC (not required for Redstone)
- Gateway Utility Request CSC
- Gateway Maintenance User Interface CSC

1.1.5 Gateway Common Services CSCI Data Flow/Interface Diagram

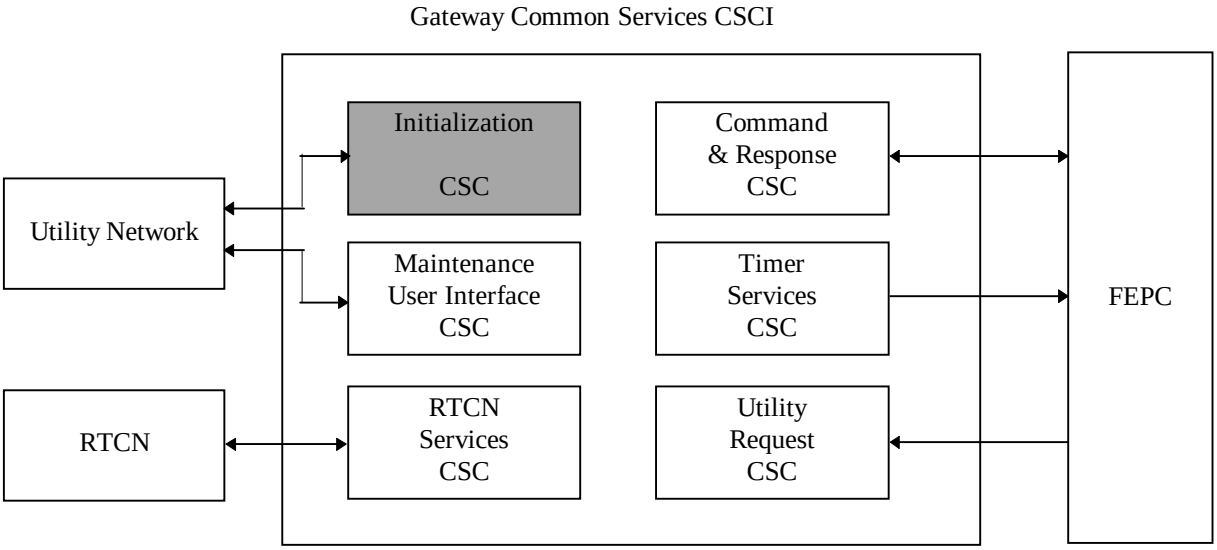


1.2 Gateway Initialization CSC

1.2.1 Gateway Initialization CSC Introduction

1.2.1.1 Gateway Initialization CSC Overview

The Gateway Initialization CSC is responsible for the initialization sequence of the Gateway. It is part of the Gateway Common Services CSCI and is resident in the GCP.



1.2.1.2 Gateway Initialization CSC Operational Description

The Gateway Initialization CSC controls the initialization sequence of the Gateway. The Gateway's initialization sequence is divided into four (4) modes of operation that allow a synchronized boot sequence between all the Single Board Computers resident in the Gateway.

1.2.2 Gateway Initialization CSC Specifications

1.2.2.1 Gateway Initialization CSC Groundrules

- SCID and TCID tables will be resident on the local hard drive.
- Initial SCID will be loaded via utility port during initial Gateway installation.
- The Gateway Initialization CSC will support the following modes of operation in order to synchronize the Gateway's boot sequence:
  - SCID Initialization
  - SCID/TCID Load
  - Ready
  - Operational
- The initialization modes interface with the rest of the system via the RTCN.
- In case of a boot failure, the Gateway will transition to the 'Primitive mode' in order to allow a new SCID to be loaded.
- The Primitive mode will support communications via the ethernet and the console port.

- The Gateway Initialization CSC will support the following initialize commands:
  - Init SCID
  - Init TCID
  - Activate Gateway
  - Terminate Gateway
- For Redstone, configuration commands such as Activate Data Acquisition and Activate Global Commands will be supported via the Utility Network.

1.2.2.2 Gateway Initialization CSC Functional Requirements

1. Gateway Initialization CSC shall support the following commands:
   Init SCID (reboot), Init TCID, Activate Gateway, and Terminate Gateway
2. Gateway Initialization CSC shall record initialization messages on local storage media.
3. No external commands shall be accepted during the SCID Initialization mode.
4. Gateway Initialization CSC shall transition to SCID/TCID Load mode when SCID software load is complete.
5. When in SCID/TCID Load mode, only Init SCID, Init TCID, and Status Info. commands shall be accepted.
6. Gateway Initialization CSC shall perform a reboot and enter SCID Initialization mode when the Init SCID command is received.
7. Gateway Initialization CSC shall transition to Ready mode when the GSE Gateway Services transition to Ready mode
8. Gateway Initialization CSC shall transition to Operational mode when the GSE Gateway Services transition to Operational mode.
9. When in Operational mode, the Init SCID, Init TCID and Activate Gateway shall not be accepted.
10. Gateway Initialization CSC shall terminate all tasks and enter SCID/TCID Load mode when the Terminate Gateway command is received.
11. Gateway Initialization CSC shall perform resource/memory de-allocation prior to returning to a previous mode.
12. Gateway Initialization CSC shall generate a system message prior to termination due to an error (post Redstone).
13. Gateway Initialization CSC shall record all termination messages on local storage media.
14. Gateway Initialization CSC shall generate a system message prior to termination (Terminate Gateway command).

1.2.2.3 Gateway Initialize/Re-initialize CSC Performance Requirements

No performance requirements have been identified for the Gateway Initialize/Re-initialize CSC for the Redstone delivery.

1.2.2.4 Gateway Initialization CSC Interfaces/Data Flow Diagram

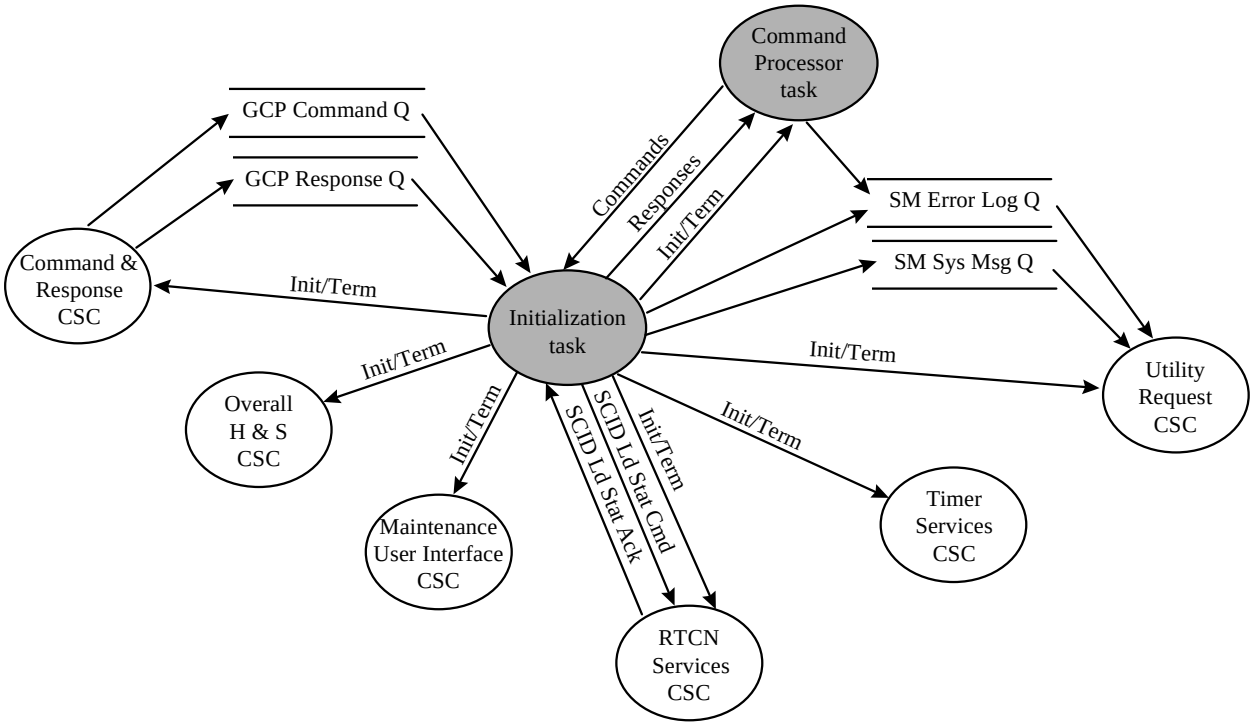
Refer to diagram in section 1.1.5.

1.2.3 Gateway Initialization CSC Design Specification

The Gateway Initialization CSC is made up of two tasks: Command Processor and Initialization. The Command Processor is the task responsible for routing GCP commands and responses. The Initialization task is responsible for

initializing all the CSC's resident on the GCP. It is also responsible for the administration of the Gateway Operating Modes (See Sec. 1.1.3).

1.2.3.1 Gateway Initialization CSC Detailed Data Flow



The Initialization task utilizes RTCN Services in order to synchronize boot with the CM Server. The Command Processor task receives commands via the Command & Response CSC. The Initialization task is responsible for initializing and terminating all other CSCs on the Gateway Control Processor (GCP). CSC initialization is controlled by calling an initialization function provided by each CSC. Error and System Messages are passed to Utility Request CSC.

1.2.3.2 Gateway Initialization CSC External Interfaces

1.2.3.2.1 Gateway Initialization CSC Message Formats

1.2.3.2.1.1 Activation successful

Message Number = _____
Message Group = _____
Severity = _____

%s Gateway Activation successful

Insert #1 = Text string Gateway Logical Name (e.g. GS1A)

1.2.3.2.1.2 Activation failed

Message Number = _____
Message Group = _____
Severity = _____

%s Gateway Activation failed

Insert #1 = Text string Gateway Logical Name (e.g. GS1A)

1.2.3.2.1.3 Termination successful

Message Number = _____
Message Group = _____
Severity = _____

%s Gateway termination successful

Insert #1 = Text string Gateway Logical Name (e.g. GS1A)

1.2.3.2.1.4 Termination failed

Message Number = _____
Message Group = _____
Severity = _____

%s Gateway termination failed

Insert #1 = Text string Gateway Logical Name (e.g. GS1A)

1.2.3.2.2 Gateway Initialization CSC Interprocess Communication

1.2.3.2.2.1 CM Server Command to Init SCID or Init TCID

CM Server Command to Initialize Gateway SCID or TCID (Routing Code 9, Request ID 1)

Bytes	C-C TO DESTINATION(S)
	Header
1	Spare
1	= 1 = CM Server Init SCID to Gateway = 2 = CM Server Init TCID to Gateway
14	SCID or TCID Name

1.2.3.2.2.2 Gateway to CM Server SCID Status Command

Gateway to CM Server SCID Init Status (Routing Code 9, Request ID 5)

Bytes	C-C TO DESTINATION(S)	Bytes	RESPONSE FROM DESTINATION
	Header		Header
1	= 0 = Successful = 1 = Unsuccessful (POST failed)		

Response Completion Codes:
0 Successful
800 Fail

1.2.3.2.2.3 CM Server Command to Activate Gateway

Activate Gateway (Routing Code 8, Request ID 1)

Bytes	C-C TO DESTINATION(S)	Bytes	RESPONSE FROM DESTINATION
	Header		Header
1	= 1 = Activate = 2 = Start Processing (Not Redstone) = 3 = Activate and Start Processing (Not Red.)		
2	MSB = 1 = HIM Type II = 0 = HIM Type I		

Response Completion Codes:
0 Successful
800 Fail

1.2.3.2.2.4 CM Server Command to Terminate Gateway

Terminate Subsystem (Routing Code 18, Request ID 1)

Bytes	C-C TO DESTINATION(S)	Bytes	RESPONSE FROM DESTINATION
	Header		Header

Response Completion Codes:
0 Successful
800 Fail

1.2.3.2.2.5 Gateway Configuration Status

Gateway Configuration Status (Routing Code 8, Request ID 24)

Bytes	C-C TO DESTINATION(S)	Bytes	RESPONSE FROM DESTINATION
	Header		Header
		1	Gateway Mode 0 = SCID initialization 1 = SCID/TCID load 2 = Ready 3 = Operational
		1-30	SCID version name (ASCII)
		1-30	TCID version name (ASCII)

Response Completion Codes:
0 Successful
800 Fail

1.2.3.2.3 Gateway Initialization CSC Table Format

1.2.3.2.3.1 Activity File

The activity file created and loaded by OPS CM as part of the gateway's TCID will have the following file name: /clcs/tcid/act_data. The contents of the file will have the following format.

Line #	Field Description	Format	Max. Length
1	Activity Name	Null-terminated ASCII string	30
2	SCID Version	Null-terminated ASCII string	30
3	TCID Version	Null-terminated ASCII string	30
4	Tail ID	Null-terminated ASCII string	8
5	Flight Number	Null-terminated ASCII string	10
6	End Item Location	Null-terminated ASCII string	8

NOTE: Will the Activity File include the Gateway Logical Name (e.g. GS1A) and the System Synchronous Rate?

1.2.3.3 Gateway Initialization CSC Test Plan

1.2.3.3.1 Environment

A development Gateway will be configured as a GSE Gateway. Each of the commands supported by the Initialization CSC for the Redstone delivery will be sent. The action taken and the response returned will be verified.

1.2.3.3.2 Test Tools

The Gateway will be commanded using a CCP simulator test tool developed by the Gateway group. This tool is capable of generating and displaying the responses of all Gateway commands supported by Redstone.

1.2.3.3.3 Test Cases

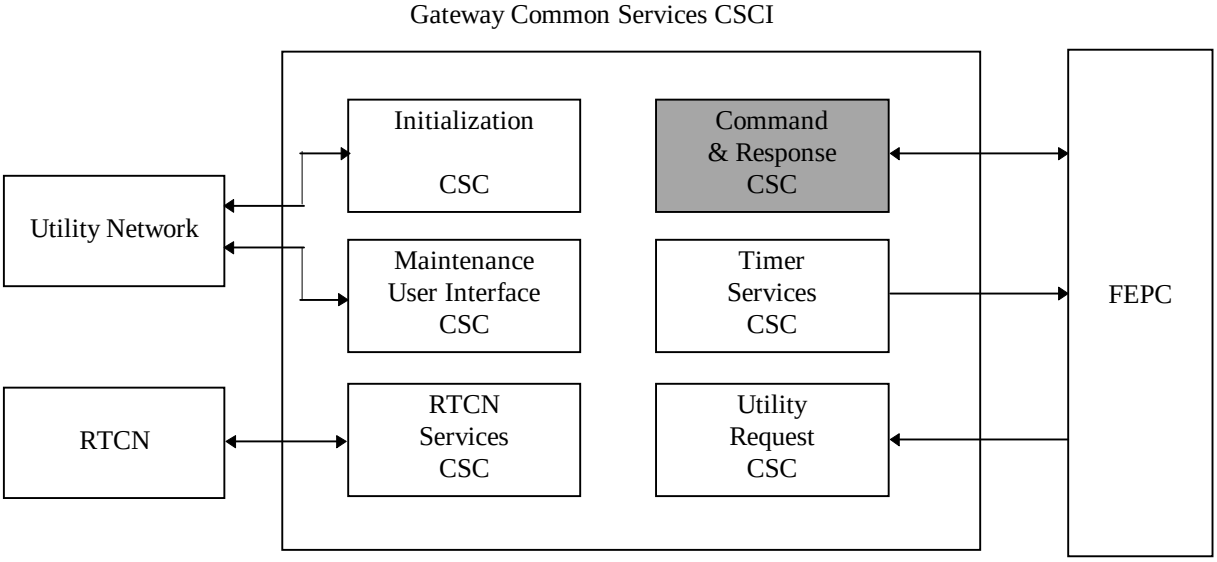
Load SCID, Load TCID, Activate Gateway, and Terminate Gateway.

1.3 Gateway Command and Response CSC

1.3.1 Gateway Command and Response CSC Introduction

The Gateway Command and Response CSC provides a communication path between the Gateway resident processors and the RTCN. The Gateway Command and Response CSC is a set of processes and tables residing on the Gateway Control Processor that conveys both command and response packets between the RTCN and the Gateway's processors.

1.3.1.1 Gateway Command and Response CSC Overview



1.3.1.2 Gateway Command and Response CSC Operational Description

During Gateway initialization, each Gateway resident processor registers the specific command and response services it requires (e.g. route codes, command and response queues). All shared message queues, a routing table, and a transaction table are built using the registered information.

An incoming RTCN command is indexed into the routing table by its route code, stored in the transaction table if a response is required, checked for command priority, and relayed to the appropriate Gateway processor(s) command queues. A Gateway processor generated response is indexed into the transaction table by transaction ID to verify a response was expected, and relayed to the RTCN.

The Gateway Command and Response CSC is also capable of relaying to the RTCN any commands issued by the Gateway resident processors, and handling the incoming responses associated with those commands. This feature will not be supported in the Redstone delivery.

1.3.2 Gateway Command and Response CSC Specifications

1.3.2.1 Gateway Command and Response CSC Groundrules

- Gateway Command and Response CSC will support an initialization shared memory message queue by which each Gateway resident processor may:
  - Register for specific Command and Response services.
  - Register its required route codes.

- Gateway Command and Response CSC will support all RTCN commands and responses including:
  - Commands received from the RTCN that need to be routed to more than one Gateway resident processor.
  - Commands that expect multiple responses (not supported in the Redstone delivery).
- Gateway Command and Response CSC will build and maintain the following tables:
  - Route Table: Maps route codes to Gateway resident processors.
  - Transaction Table: Tracks commands and their associated required responses.
- Gateway Command and Response CSC communication will be implemented as follows:
  - Communication with the Gateway resident processors is by shared memory message queues.
  - Communication with the RTCN through the ATM Network Adapter Board is by the API provided by the Network Services CSCI.
- All Gateway generated responses will be time stamped immediately prior to transfer to the RTCN. The time stamp will reflect the time the response left the gateway, not when processing occurred.

1.3.2.2 Gateway Command and Response CSC Functional Requirements

The Functional Requirements for Gateway Command and Response CSC are arranged in the following major/minor functions:

1. Initialization
2. Command Processing
3. Response Packet Building

I. Initialization

A. Gateway Command and Response CSC shall provide an initialization shared memory message queue for Gateway resident processor or GCP resident process registration.

B. Gateway Command and Response CSC shall provide the following services to each Gateway resident processor and to GCP resident processes:

1. High and normal priority receive command queues.
2. Generate response queue.
3. Change data queue.
4. High and normal priority generate command queue(not supported in the Redstone delivery).
5. Receive response queue (not supported in the Redstone delivery).
6. Route code registration.

C. Registration for any command queues shall result in the creation of both a high and normal priority queues.

D. Gateway Command and Response CSC shall build and maintain a Route Table which will contain registered routes for each Gateway resident processor and GCP resident processes.

E. Gateway Command and Response CSC shall build and maintain a Transaction Table which will track responses during command processing.

F. For each requested response queue, Gateway Command and Response CSC shall spawn a task to monitor activity on the queue.

II. Command Processing

A. Gateway Command and Response shall receive commands asynchronously from the ATM Network Adapter Board using the API provided by Network Services CSCI.

B. When an incoming command's route code is not found in the Route Table, a NACK will be returned to the command's source.

C. When an incoming command's route code is found in the Route Table, the command shall be forwarded to each processor/process that registered for that route code.

D. When an incoming command has been designated as high priority, the command shall be placed in the appropriate processor(s)/process(es) high priority receive command queue.

E. When a received command expects a response, the command's header shall be copied into the Transaction Table for reference during Response Packet Building.

III. Response Packet Building

A. Gateway Command and Response CSC shall build and send response packets to the ATM Network Adapter Board using the API provided by the Network Services CSCI.

B. For each generate response queue, Gateway Command and Response CSC shall have a dedicated process to monitor queue activity.

C. Each generate response queue monitor shall read asynchronously from the shared memory message queues.

D. When a generated response is read from the shared memory message queues and not found in the Transaction Table, a System Message will be issued and TBD.

E. When a generated response is read from the shared memory message queues and found in the Transaction Table, Gateway Command and Response CSC shall time stamp the packet and write it to the RTCN.

F. When the generated response is one of several expected responses, the response shall be stored in the Transaction Table until all expected responses have been generated. At that time, Gateway Command and Response CSC shall time stamp and send the response to the RTCN (not supported in the Redstone delivery).

G. When a routed command does not return a response within TBD, a Response packet shall be returned to the CCP with a COMMAND TIMED OUT code.

1.3.2.3 Gateway Command and Response CSC Performance Requirements

I. Processing Speed

A. Gateway Command and Response CSC shall be capable of processing 500 commands per second.

II. Other

A. Gateway Command and Response CSC shall generate a system message prior to termination due to an error (post Redstone)

Note: No other Gateway Command and Response CSC performance requirements have been identified.

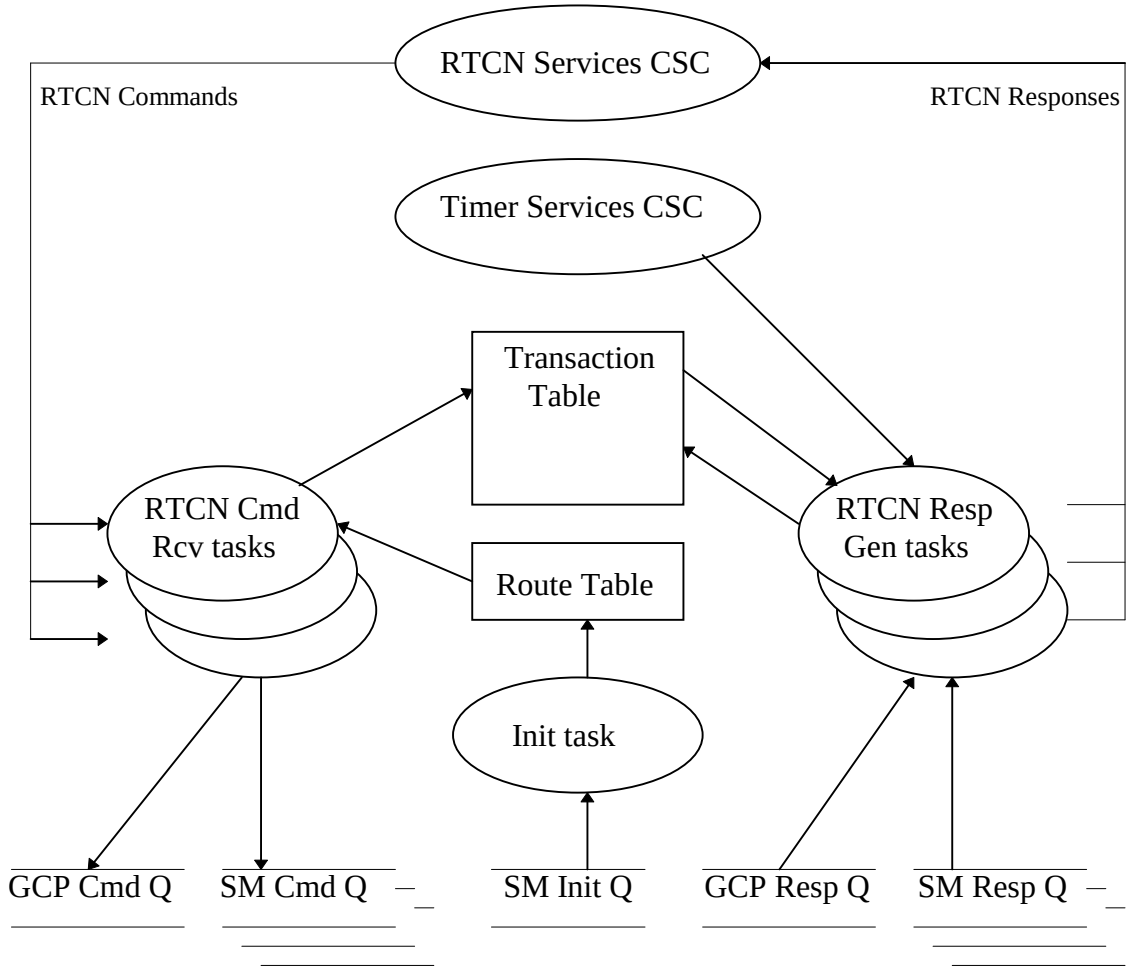
1.3.2.4 Gateway Command and Response CSC Interfaces/Data Flow Diagram

Refer to diagram in section 1.1.5.

1.3.3 Gateway Command and Response CSC Design Specification

The GCP Command and Response CSC is responsible for routing and tracking Gateway commands and responses. Received RTCN Commands are routed to the appropriate Gateway resident processor through the Command Shared Memory Message Queues. RTCN Responses are received from the Gateway resident processors through Response Shared Memory Message Queues and from this, an RTCN response is generated. Commands and responses are tracked while being relayed to insure their timely delivery.

1.3.3.1 Gateway Command and Response CSC Detailed Data Flow



GCP Command and Response is comprised of two tables and several tasks. The Initialization (Init) task creates and maintains the Initialization Shared Memory Queue. This Queue is used by the other processors in the Gateway to register for Shared Memory Queues and Route Codes. When a processor registers, it designates which Route Codes it wishes to receive, and which Queue services it requires (Command, Response, or Changed Data). The Init task is responsible for entering the Route Codes into the Route Table, and creating the requested Shared Memory Message Queues.

Incoming commands are received from the RTCN Services CSC by one of the RTCN Command Receive (RTCN Cmd Rcv) tasks. There is one RTCN Cmd Rcv task for each CCP communicating with the Gateway. The RTCN Cmd Rcv task indexes the Route Table to determine to which Queue(s) to route the incoming command. The command is stored in the Transaction Table for future reference, time-tagged using the Timer Services CSC, and written to the appropriate command Queue(s).

Gateway processor responses are received through the RTCN Response Generator (RTCN Resp Gen) tasks. There is an RTCN Resp Gen task for each Response Shared Memory Queue created during initialization. The response is indexed into the Transaction Table where the stored command is located. Using the response and the command, an RTCN response is built, time-tagged using the Timer Services CSC, and sent to the RTCN Services CSC for transmission to the RTCN.

1.3.3.2 Gateway Command and Response CSC External Interfaces

1.3.3.2.1 Gateway Command and Response CSC External Interface Calls

1.3.3.2.1.1 GCP Services API Initialization

STATUS gcps_api_init(int services,
int num_route_codes,
ROUTE_CODE_PTR route_code_list);

Parameters: services — A bit pattern with three bits defined as:
GCPS_RECEIVE_COMMAND
GCPS_SEND_COMMAND
GCPS_CHANGE_DATA
Any or all bits may be specified simultaneously or the following may be used to request all services:
GCPS_ALL_SERVICES

num_route_codes — The number of command route codes this processor receives. Zero if the receive command service is not required.

route_code_list — A pointer to a list of the command route codes this processor receives. NULL if the receive command service is not required.

Returns: OK or ERROR as defined in vxWorks.h

Description: Performs all initialization routines required by the GCP Services API.

1.3.3.2.1.2 Copy Header

void gcps_copy_header(GCPS_COMMAND_INFO_PTR from,
GCPS_COMMAND_INFO_PTR to);

Parameters: from, to — pointers to GCPS_COMMAND_INFO_TYPE structures. The <from> structure will be copied into the <to> structure.

Returns: none

Description: Copies data from one GCPS_COMMAND_INFO_TYPE structure to another.

1.3.3.2.1.3 Wait Command

STATUS gcps_wait_command(GCPS_COMMAND_INFO_PTR info,
void *buffer);

Parameters: header — pointer to a GCPS_COMMAND_INFO_TYPE structure which is filled in as part of the call.

buffer — pointer to a buffer to be filled in with the command payload body

Returns: OK or ERROR as defined in vxWorks.h

Description: This routine will pend on the gcps_rcv_cmd_ready_sem semaphore then use gcps_get_priority() to determine which queue has data and gcps_get_command() to read the data. The command information and data is returned to the caller.

1.3.3.2.1.4 Get Priority

int gcps_get_priority();

Parameters: none

Returns: GCPS_NO_COMMAND, GCPS_NORMAL_PRIORITY or GCPS_HIGH_PRIORITY

Description: Checks each of the incoming command queues and returns an indication of which queue has commands. The high priority queue will be tested first. If no command is found, the normal priority queue is tested. The return from this function may be used as the priority input parameter in the gcps_get_command() function.

1.3.3.2.1.5 Get Command

STATUS gcps_get_command(GCPS_COMMAND_INFO_PTR info,
void *buffer, int priority);

Parameters:
- header — pointer to a GCPS_COMMAND_INFO_TYPE structure which is filled in as part of the call
- buffer — pointer to a buffer to be filled in with the command data
- priority — GCPS_NORMAL_PRIORITY or GCPS_HIGH_PRIORITY

Returns: OK or ERROR as defined in vxWorks.h

Description: Reads an incoming command from the specified command queue. Fills in a GCPS_COMMAND_INFO_TYPE structure and then copies the command data into the data buffer. The command processor should use the route code and request id to determine what command to execute, perform the command, then call gcps_generate_response() with the appropriate response data for the command.

1.3.3.2.1.6 Generate Response

STATUS gcps_generate_response(GCPS_COMMAND_INFO_PTR info,
void *buffer);

Parameters:
- header — pointer to the GCPS_COMMAND_INFO_TYPE structure which was returned from the gcps_wait_command() or gcps_get_command() call. The size_in_bytes parameter in this structure must be modified to reflect the size of the response data. The completion_code parameter must be updated with the proper status. The remainder of the structure must be left unmodified.
- buffer — pointer to a buffer which contains the response data.

Returns: OK or ERROR as defined in vxWorks.h

Description: Generates a response message to the subsystem which originated the command contained in the command header structure. This function will also handle generating the time tag for the response.

1.3.3.3 Gateway Command and Response CSC Test Plan

1.3.3.3.1 Environment

A development Gateway will be configured as a GSE Gateway. RTCN commands will be sent to the development Gateway. The action taken and the responses returned will be verified.

1.3.3.3.2 Test Tools

The Gateway will be commanded using a CCP simulator test tool developed by the Gateway group. This tool is capable of generating and displaying the responses of all Gateway commands supported by Redstone.

1.3.3.3.3 Test Cases

1. Gateway Process/Processor registration via the Initialization Queue.
2. Routing of Commands to the GCP local processes and receiving the expected responses.
3. Routing of Commands to the Gateway resident processors and receiving the expected responses

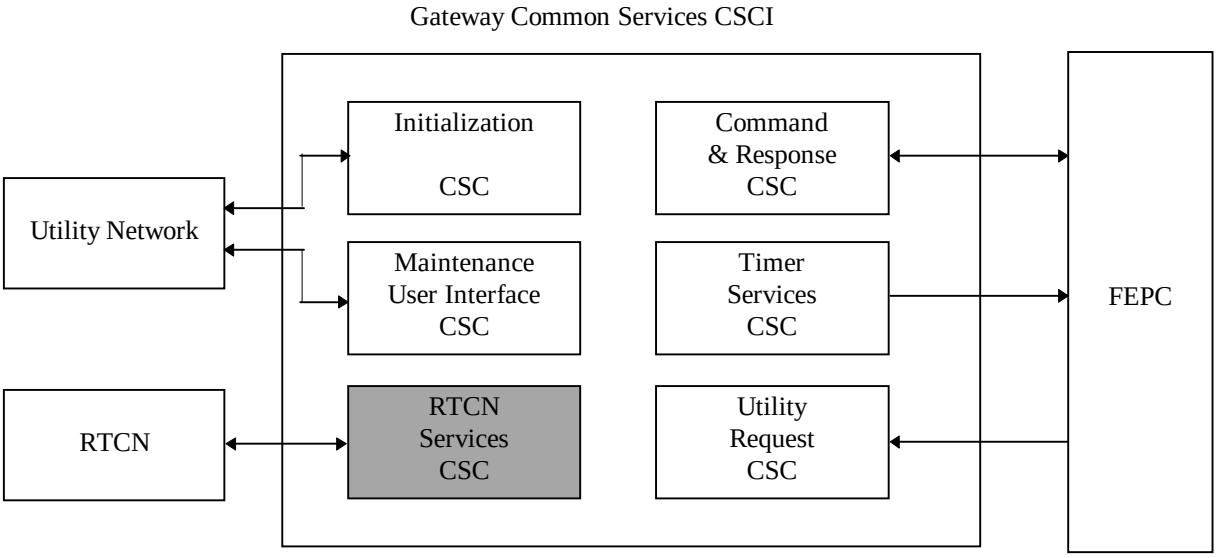
NOTE: Requirement 1.3.2.3 I. Processing Speed A. will be tested by the GSE Services CSCI.

1.4 Gateway RTCN Services CSC

1.4.1 Gateway RTCN Services CSC Introduction

1.4.1.1 Gateway RTCN Services CSC Overview

The Gateway RTCN Services CSC is responsible for the different services needed by the Gateway when interfacing with the RTCN. It is part of the Gateway Common Services CSCI and is resident in the GCP.



1.4.1.2 Gateway RTCN Services CSC Operational Description

The Gateway RTCN Services CSC contains the API provided by the Network Services CSCI which is used by the Gateway components when interfacing with the RTCN. Also, when interrupted by the Timer Services CSC, the Gateway RTCN Services CSC will build RTCN change data packets and transfer them to the RTCN. These RTCN change data packets contain change measurements provided by the FEPC.

1.4.2 Gateway RTCN Services CSC Specifications

1.4.2.1 Gateway RTCN Services CSC Groundrules

- The Network Services CSCI API library will be linked as part of the Gateway SCID.
- Timer Services CSC will interrupt RTCN Services CSC at the system synchronous rate.
- The system synchronous rate will be provided as part of the TBD. For Redstone, the system synchronous rate will be 10 milliseconds by default.
- The FEPC will provide change measurements via the SM Change Data Queue.
- Change measurements are within the system synchronous rate period.

1.4.2.2 Gateway RTCN Services CSC Functional Requirements

1. Gateway RTCN Services CSC shall generate a system message prior to termination due to an error (post Redstone).

2. The Gateway RTCN Services CSC shall provide change data packets to the RTCN at the system synchronous rate.

1.4.2.3 Gateway RTCN Services CSC Performance Requirements

1. The Gateway RTCN Services CSC shall provide change data packets to the RTCN at the system synchronous rate.

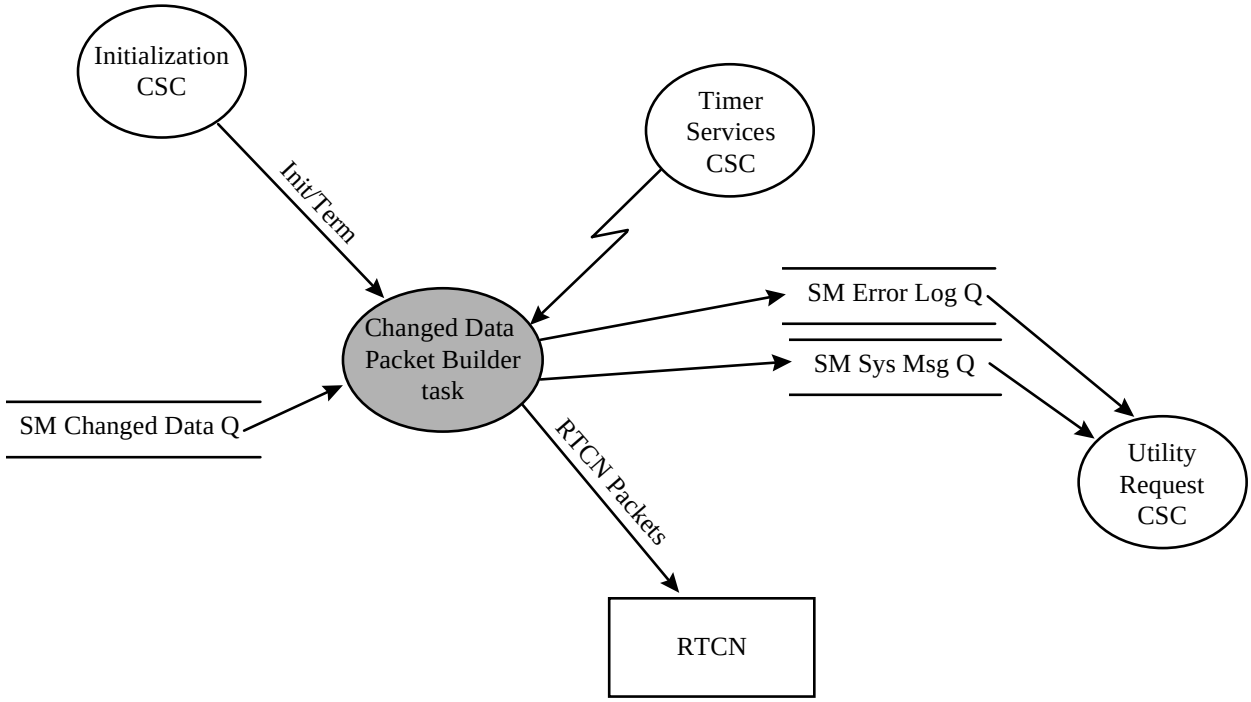
1.4.2.4 Gateway RTCN Services CSC Interfaces/Data Flow Diagram

Refer to diagram in section 1.1.5.

1.4.3 Gateway RTCN Services CSC Design Specification

The Gateway RTCN Services CSC contains the APIs provided by the Network Services CSC which is used by the Gateway components when interfacing with the RTCN. The Gateway RTCN Services CSC also provides the Change Data Packet Builder which builds and transmits the change data via RTCN. The changed data is provided by the FEPC. This transmission is done at the system synchronous rate.

1.4.3.1 Gateway RTCN Services CSC Detailed Data Flow



The Changed Data Packet Builder task is interrupted by the Timer Services CSC at the system synchronous rate. It then utilizes RTCN Services in order send packets via the RTCN. Changed Data is queued at the SM Changed Data Queue by the FEPC. The Changed Data Packet Builder task is initialized/terminated by the Init/Re-Init CSC. Error and System Messages are passed to Utility Request CSC via the SM Error and Sys. Msg. queues.

1.4.3.2 Gateway RTCN Services CSC External Interfaces

1.4.3.2.1 Gateway RTCN Services CSC External Interface Calls

1.4.3.2.1.1 Network Services APIs as defined in the Network Services Interface Definition Document

1.4.3.2.1.2 Write Change Data

```
STATUS gcps_write_change_data( UNSIGNED16 status,
                               UNSIGNED32 fdid,
```

int data_size,
void *data);

Parameters:	status	Measurement status (FD processing active / inhibited, FD warning / no warning)
	fdid	Function Designator ID from the measurement data table.
	data_size	size of data element in bytes
	data	pointer to data element
Returns:	OK or ERROR as defined in vxWorks.h	
Description:	Outputs change data for placement in the change data packet.	

1.4.3.2.2 Gateway RTCN Services CSC Table Format

1.4.3.2.2.1 Static Address Table

Each communications path to/from the Gateway is named in this file and a mapping from this name to the multicast address/port number for data and acknowledgments associated with this path is provided. The proposed format for each entry is as follows:

<stream_name> <data_ip_address> <data_udp_port> <ack_ip_address> <ack_udp_port>

where:
stream_name is the name of the data stream,
data_ip_address is the address of the data stream (broadcast or multicast),
data_udp_port is the UDP port number associated with the data stream,
ack_ip_address is the address to which ack's are sent by the receiver,
ack_udp_port is the UDP port to which ack's are sent by the receiver.

This table is currently generated by hand. It must be present for on the Gateway.

1.4.3.3 Gateway RTCN Services CSC Test Plan

1.4.3.3.1 Environment

A development Gateway will be configured as a GSE Gateway.

1.4.3.3.2 Test Tools

The Gateway will be commanded using a CCP simulator test tool developed by the Gateway group.

1.4.3.3.3 Test Cases

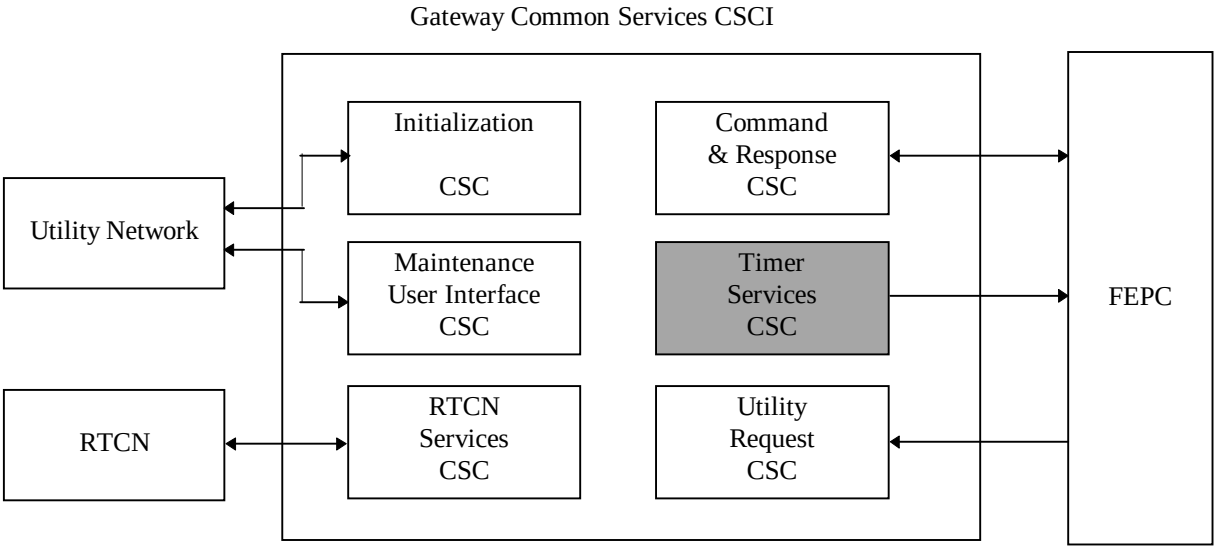
While Data Acquisition is active, verify that the Gateway is generating changed data packets at the system synchronous rate. This test case will also test a portion of the Gateway Timer Services CSC.

1.5 Gateway Timer Services CSC

1.5.1 Gateway Timer Services CSC Introduction

1.5.1.1 Gateway Timer Services CSC Overview

The Gateway Timer Services CSC is responsible for the different services needed by the Gateway when interfacing with the time interface card. It is part of the Gateway Common Services CSCI and is resident in the GCP.



1.5.1.2 Gateway Timer Services CSC Operational Description

The Gateway Timer Services CSC is responsible for providing the initialization routine for the time interface card. The Gateway Timer Services CSC will also provide services to read the time as provided by the time interface card.

1.5.2 Gateway Timer Services CSC Specifications

1.5.2.1 Gateway Timer Services CSC Groundrules

- Gateway Timer Services CSC will provide the initialization routine for the time interface card.
- Gateway Timer Services CSC will provide common services as part of the GCP Services API.
- GCP Services API will include:
  - Get time in BCD with micro-second resolution.
  - Get millisecond time of day (32-bit integer).
  - Get Julian time of year, JTOY (32-bit integer)
- The time interface card will use an external IRIG-B signal in order to synchronize the time of day.

1.5.2.2 Gateway Timer Services CSC Functional Requirements

1. Gateway Timer Services CSC will interrupt the RTCN Services' Change Data Packet Builder at the system synchronous rate.
2. Gateway Timer Services CSC shall generate a system message prior to termination due to an error (post Redstone).

1.5.2.3 Gateway Timer Services CSC Performance Requirements

1. Gateway Timer Services CSC API will return time within 10 μ seconds from the time of the call.

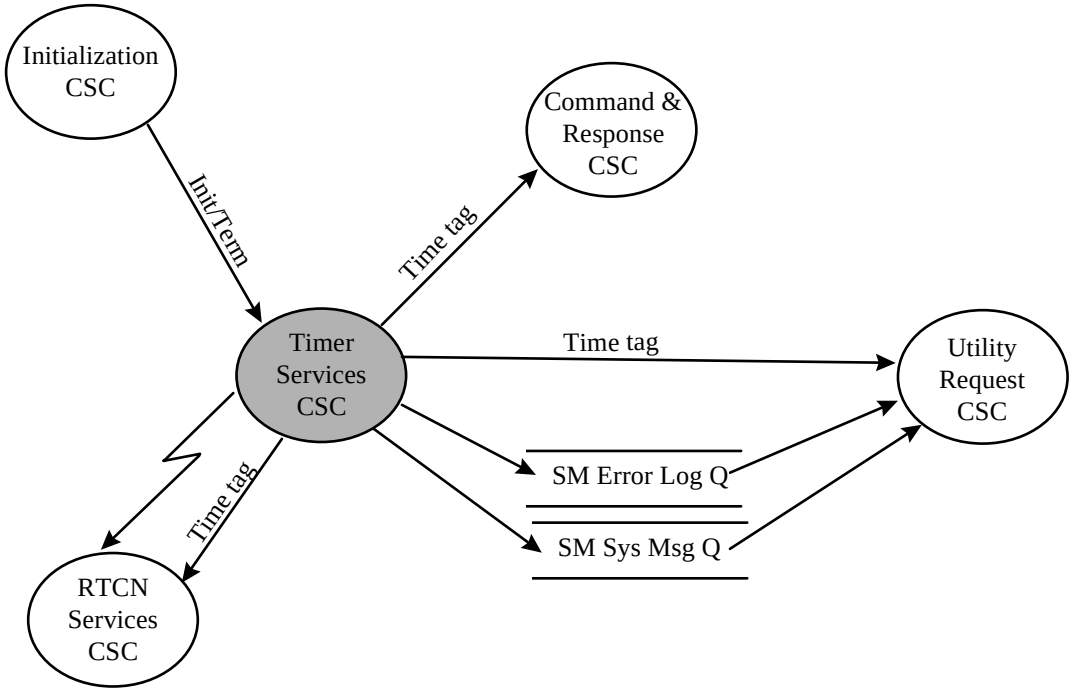
1.5.2.4 Gateway Timer Services CSC Interfaces/Data Flow Diagram

Refer to diagram in section 1.1.5.

1.5.3 Gateway Timer Services CSC Design Specification

The Gateway Timer Services CSC provides the APIs used to read the time used for tagging. It is also responsible for interrupting the RTCN Services CSC at the system synchronous rate.

1.5.3.1 Gateway Timer Services CSC Detailed Data Flow



The Gateway Timer Services CSC is responsible for interrupting the RTCN Services CSC at the system synchronous rate. It also provides to all the CSCs in the Gateway the time used for tagging packets, error log entries, etc.

1.5.3.2 Gateway Timer Services CSC External Interfaces

1.5.3.2.1 Gateway Timer Services CSC External Interface Calls

1.5.3.2.1.1 Get Time

STATUS gcps_get_time(GCPS_TIME_PTR time);

Parameters: time — a pointer to a GCPS_TIME_TYPE structure which will be filled in by the call.

Returns: OK or ERROR as defined in vxWorks.h

Description: Reads the time card and returns time in a GCPS_TIME_TYPE structure.

1.5.3.2.1.2 Get ASCII Time

char *gcps_get_ascii_time(char *return_string, char *format_string)

Parameters:	return_string	Pointer to a character string into which the formatted time text will be placed
	format_string	Pointer to a character string containing format codes.
Returns	Pointer to return_string	

Description: Reads current time and generates an ascii character representation of time based on a caller provided format string

Valid format codes are:

%n	insert a new-line character
%t	insert a tab character
%m	month of year - 01 to 12
%d	day of month - 01 to 31
%y	2 digit year - 00 to 99
%Y	4 digit year - 0000 to 9999
%D	date as mm/dd/yy
%H	hour - 00 to 23
%M	minute - 00 to 59
%S	second - 00 to 59
%T	24 hour time as HH:MM:SS.mmm
%j	day of year - 001 to 366
%w	day of week - Sunday = 0
%a	abbreviated weekday - Sun to Sat
%h	abbreviated month - Jan to Dec
%r	12 hour time as HH:MM:SS.mmm XM
%u	microseconds - 0 to 999

Examples: Format string is "%h. %d, %Y"
Return string is "Feb. 04, 1997"

Format string is "%j/%T"
Return string is "035/15:26:32.059"

NOTE: Desired format: 035:1526/32.059

1.5.3.3 Gateway Timer Services CSC Test Plan

The Gateway Timer Services CSC interrupts will be tested by the Gateway RTCN Services CSC Test Plan. The Gateway Timer Services CSC APIs will be tested by the Gateway Utility Request CSC Test Plan.

NOTE: Test Requirement #1 in 1.6.2.3

1.6 Gateway Overall Health & Status CSC

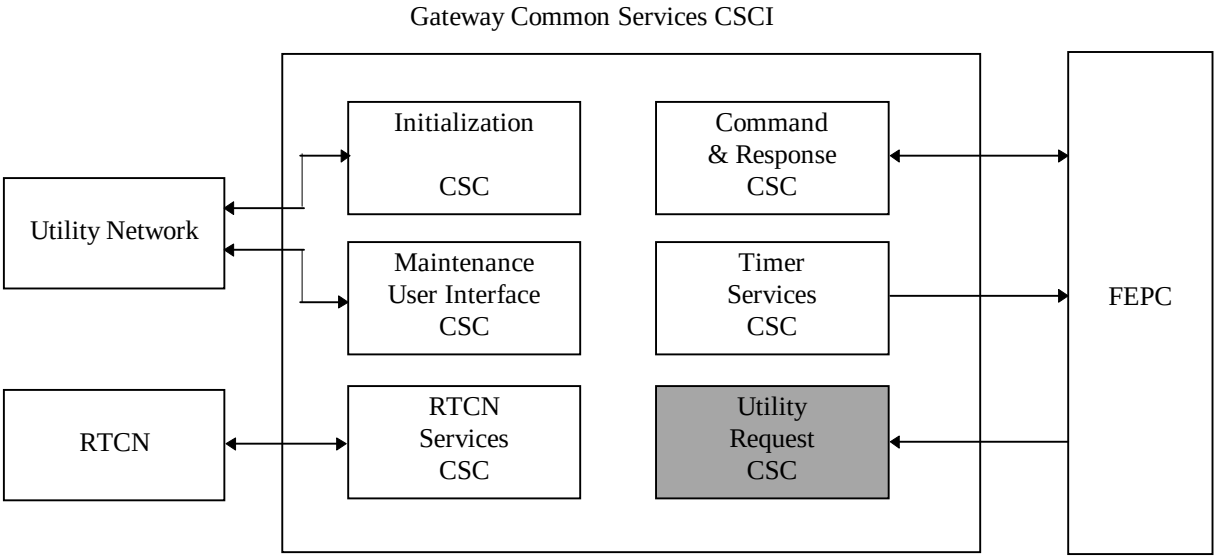
- Gateway Overall Health & Status CSC is not part of the Redstone baseline.

1.7 Gateway Utility Request CSC

1.7.1 Gateway Utility Request CSC Introduction

1.7.1.1 Gateway Utility Request CSC Overview

The Gateway Utility Request CSC is responsible for transmitting System Messages, error logging and status logging. It is part of the Gateway Common Services CSCI and is resident in the GCP.



1.7.1.2 Gateway Utility Request CSC Operational Description

Gateway Utility Request CSC will provide the capability for any resource in the Gateway to generate System Messages via RTCN. It will also provide the capability to log error and status messages to local disk. These error and status message can also be routed to the console port.

1.7.2 Gateway Utility Request CSC Specifications

1.7.2.1 Gateway Utility Request CSC Groundrules

- Gateway Utility Request CSC will provide an API that will allow any resource in the Gateway to generate System Messages, log to local disk error/status messages, and route error/status messages to the console port.
- The message number and the message structure used to format System Messages will be defined in a header file that is used by the GCP and by the CLCS CCP to format the text output.
- This header file needs to be provided to the Gateway development team so that it can be included as part of the Gateway SCID build.
- System Message Services CSC will provide the System Message API.

1.7.2.2 Gateway Utility Request CSC Functional Requirements

1. Gateway Utility Request CSC shall provide a method for all resources in the Gateway to send System Messages via RTCN.
2. Gateway Utility Request CSC API shall return success or failure status back to calling CSC.
3. Gateway Utility Request CSC API shall provide the capability for specifying message parameters.
4. Gateway Utility Request CSC shall support a minimum of TBD message severity level classifications.

5. Gateway Utility Request CSC shall define the Packet Payload content which includes an indicator specifying message is to be logged by SDC.
6. Gateway Utility Request CSC shall distribute the message to multiple RTPS destinations and to the SDC for recording.

1.7.2.3 Gateway Utility Request CSC Performance Requirements

No performance requirements have been identified for the Gateway Utility Request CSC for the Redstone delivery.

1.7.2.4 Gateway Utility Request CSC Interfaces/Data Flow Diagram

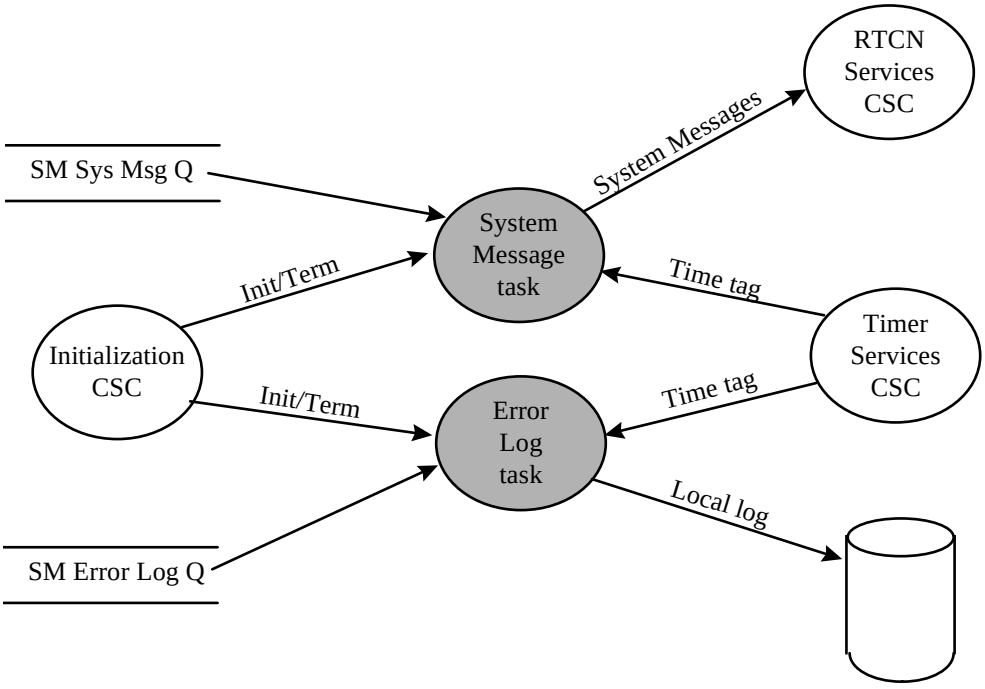
Refer to diagram in section 1.1.5.

1.7.3 Gateway Utility Request CSC Design Specification

The Gateway Utility Request CSC is made up of two tasks: Error Log task and System Message task. Each task monitors its respective queue and transfers the queue entries to either the Local Log or the RTCN (System Messages).

Also, the Gateway Utility Request CSC provides the APIs used to log messages to the Gateway's local log and generate system messages.

1.7.3.1 Gateway Utility Request CSC Detailed Data Flow



The Gateway Utility Request CSC tasks monitor the Error Log Queue and the System Message Queue. When entries are found in either queue, the Gateway Utility Request CSC de-queues them. Based on the queue type, the entries are either logged locally or a System Message is generated.

1.7.3.2 Gateway Utility Request CSC External Interfaces

1.7.3.2.1 Gateway Utility Request CSC External Interface Calls

1.7.3.2.1.1 Log Message

STATUS gcps_log_message(int msg_type,
char *format,
[,arg]…);

Parameters:	msg_type	Identifies the message type. A bit pattern defined in the header file gcps_services_api.h as: GCPS_DISPLAY_MSG_TYPE GCPS_DISK_MSG_TYPE
	format	printf compatible text format string
	[arg]…	printf compatible argument list
Returns:	OK or ERROR as defined in vxWorks.h	
Description:	Passes a text message to the GCP which be logged to the console port and/or the local disk. The text message must be less than GCPS_LOG_MAX_MESSAGE_LENGTH after formatting (defined in gcps_services_api.h).	

1.7.3.2.1.2 System Message

Provided by System Message Services CSC.

1.7.3.3 Gateway Utility Request CSC Test Plan

1.7.3.3.1 Environment

A development Gateway will be configured as a GSE Gateway.

1.7.3.3.2 Test Tools

Connect to the Gateway via the Utility Network (which will test the Gateway Maintenance User Interface CSC). This interface will allow visibility of the Gateway Error Log. A CLCS CCP will be required to view the System Messages.

1.7.3.3.3 Test Cases

For all test cases, verify the contents of the System Message and the Error Log to make sure the time tag is correct.

1.7.3.3.3.1 Success Messages

1) Issue Activate Gateway command and verify the respective message.
2) Issue Terminate Gateway command and verify the respective message.

1.7.3.3.3.2 Fail Messages

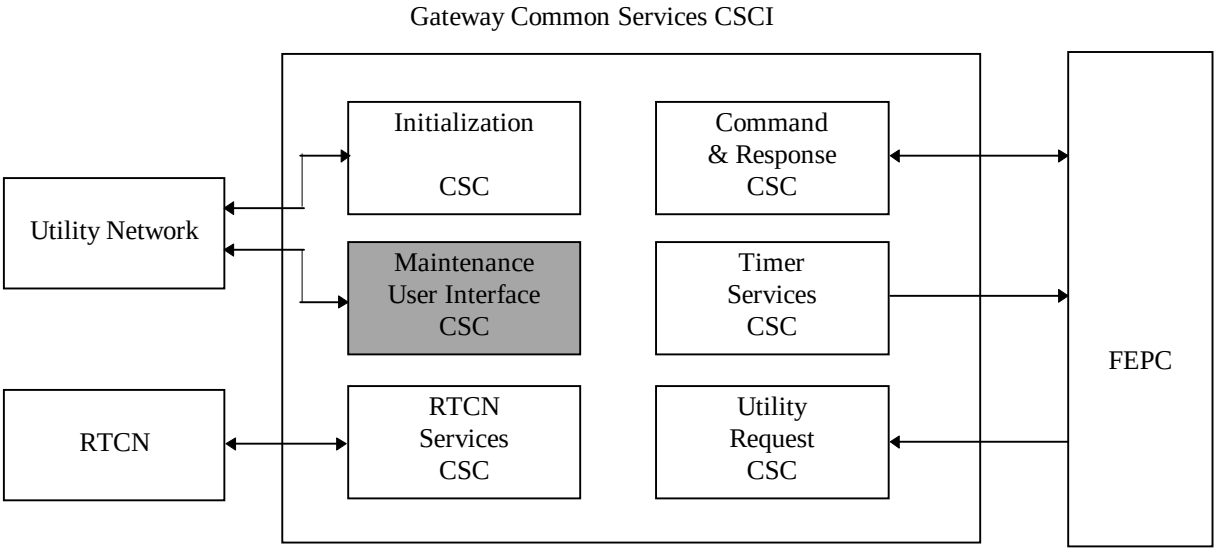
1) Issue Activate Gateway command and force the Gateway to fail by resetting the FEPC while the command is being processed. Verify the respective messages.
2) Issue Terminate Gateway command and force the Gateway to fail by resetting the FEPC while the command is being processed. Verify the respective messages.

1.8 Gateway Maintenance User Interface CSC

1.8.1 Gateway Maintenance User Interface CSC Introduction

1.8.1.1 Gateway Maintenance User Interface CSC Overview

The Gateway Maintenance User Interface CSC allows access to the Gateway via the Utility Network. It is part of the Gateway Common Services CSCI and is resident in the GCP.



1.8.1.2 Gateway Maintenance User Interface CSC Operational Description

Gateway Maintenance User Interface CSC will allow access to the Gateway via the Utility Network (ethernet and RS-232) in order to load the initial SCID during Gateway installation. It will also allow access to Health and Status information via the Utility Network.

1.8.2 Gateway Maintenance User Interface CSC Specifications

1.8.2.1 Gateway Maintenance User Interface CSC Groundrules

- For Redstone, configuration commands such as Activate Data Acquisition and Activate Global Commands will be supported via the Utility Network.
- Gateway Maintenance User Interface CSC provides the Utility Network (ethernet and console port) access to the Gateway during installation, development, and operation.
- During the Gateway's installation, the initial SCID will be loaded using the Utility Network.
- If the Gateway transitions to the 'Primitive mode', the Utility Network will be the only means of communication with the Gateway.
- Gateway Health and Status information can be accessed via the Utility Network.

1.8.2.2 Gateway Maintenance User Interface CSC Functional Requirements

1. Gateway Maintenance User Interface CSC shall generate a system message prior to termination due to an error (post Redstone).

2. Gateway Maintenance User Interface CSC shall provide access to the Gateway via the Utility Network in order to support SCID load.

3. Gateway Maintenance User Interface CSC shall provide access to the Gateway via the Utility Network when the Gateway is in the ‘Primitive mode’.

1.8.2.3 Gateway Maintenance User Interface CSC Performance Requirements

No performance requirements have been identified for the Gateway Maintenance User Interface CSC for the Redstone delivery.

1.8.2.4 Gateway Maintenance User Interface CSC Interfaces/Data Flow Diagram

Refer to diagram in section 1.1.5.

1.8.3 Gateway Maintenance User Interface CSC Design Specification

The Gateway Maintenance User Interface CSC is responsible for configuring the ethernet and console ports on the Gateway. These ports are used during Gateway installation, Gateway development, and in case the Gateway fails to boot and it transitions to the Primitive mode.

For Redstone, all the Gateway Maintenance User Interface CSC capabilities are provided by the VxWorks kernel.

1.8.3.1 Gateway Maintenance User Interface CSC Detailed Data Flow

Not applicable for Redstone.

1.8.3.2 Gateway Maintenance User Interface CSC External Interfaces

None for Redstone

1.8.3.3 Gateway Maintenance User Interface CSC Test Plan

The Gateway Maintenance User Interface CSC will be tested by the Gateway Utility Request CSC Test Plan.